

# Recovering High Dynamic Range Radiance Maps from Photographs

Christoph Gladisch

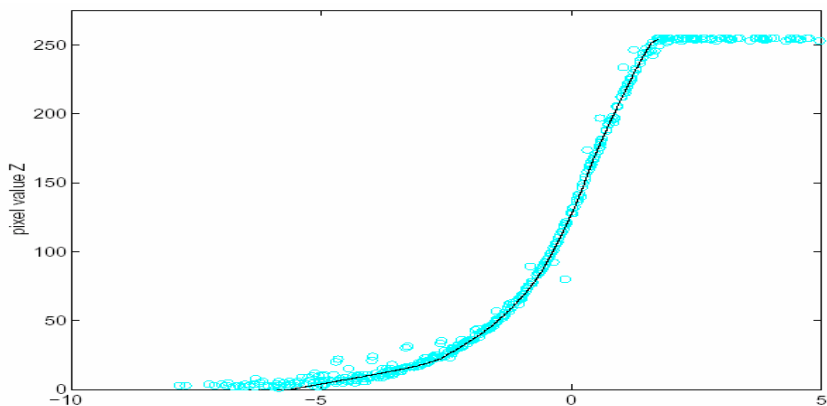
## Einführung

- Wenn eine Szene fotografiert wird, dann entsprechen die Pixelwerte nicht den relativen Leuchtdichten in der Szene.
- Es findet eine nicht lineare Abbildung der der Beleuchtung auf die Pixelwerte statt.
- CCD Kameras mappen auch nicht linear. Die Spannung eines CCD Elementes ist zwar proportional zur Beleuchtungsstärke, aber in der Kamera findet zwischen CCD chip und den Ausgabewerten eine nicht lineare Abbildung statt.

## Wertebereich kann nicht abgebildet werden.



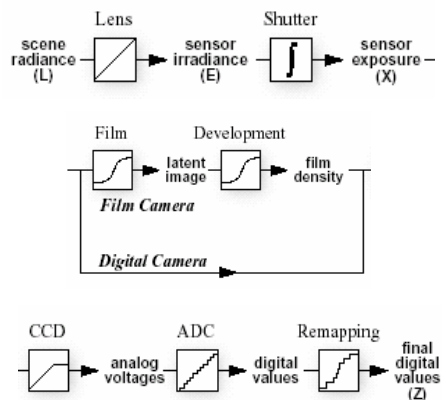
## Response Function



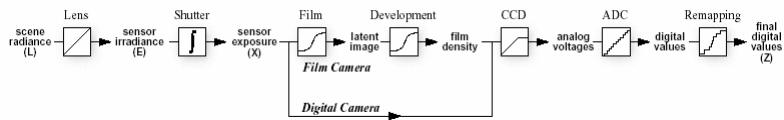
## Response Function ermitteln

- Diese nicht lineare Abbildung kann von vorn herein nicht angegeben werden, weil es sich um mehrere nicht lineare Abbildungen im ganzen Entwicklungsprozess handelt.
- Das Verhalten eines Films auf die Beleuchtungsstärke wird die charakteristische Kurve genannt

## Bildentwicklungsprozess



# Bildentwicklungsprozess



## Ziel des Verfahrens

- Mit diesem Verfahren, kann aus mehreren Bildern einer Einstellung, aber unterschiedlichen Beleuchtungszeiten, die nicht lineare response function ermitteln und die durch den Saturationspunkt abgeschnittenen Kurven der Bilder zusammenfügen.

## Saturationspunkt

- Das größte Problem der Abbildungsfunktion ist am Saturationspunkt. Das ist der maximale Wert der im Film oder von der Kamera unterschieden werden kann. Jede stärkere Beleuchtung, wird auf diesen maximalen Wert abgebildet.
- Beispiel: Das filmen(fotografieren) eines Fensters aus einem Raum heraus.

## Anwendungen

## Bildbasierte modellierung und Rendering.

Bisher wurde von der falschen Annahme ausgegangen, dass Bilder von einer Umgebung mit den selben Belichtungseinstellungen gemacht sind.

- Mit der hier vorgestellten Methode kann man Bilder die mit unterschiedlichen Belichtungseinstellungen gemacht wurden wieder zusammen kombinieren.
- Eine wichtige Anwendung dabei ist das ermitteln der BRDF

## Bildverarbeitung

Blurring, Kantendetektion, Farbkorrektur und Bildvergleiche erwarten Pixelwerte, die proportional zur Beleuchtung in der Szene sind.

- Hier Beispiel Blurring.

## Synthetisches Motionblur vs. echtem Motionblur



## HRD Motionblur vs. echtem Motionblur



## Bildkomposition

- Mischen von Bildelementen, die durch unterschiedliche Prozesse entstanden sind wie Photographi, Rendering, Bluescreen.

## Der Algorithmus

- Zuerst wird die response function ermittelt, also die Abbildung der realen Beleuchtungsstärke auf Pixelwerte.
- Mit Hilfe der response function kann dann aus mehreren Low Dynamic Range Bildern und den zugehörigen Belichtungszeiten ein High Dynamic Range Bild erzeugt werden.

## Anmerkung

- Der Begriff Beleuchtungsstärke ist nicht wirklich korrekt, weil man die spektrale Bestrahlungsstärke mit der V-Lambda Kurve multiplizieren und integrieren müsste.

## Berechnung der nicht linearen Abbildung

$$f^{-1}(Z_{ij}) = E_i \Delta t_j$$

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

$$\mathcal{O} = \sum_{i=1}^N \sum_{j=1}^P [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2$$

$$g''(z) = g(z-1) - 2g(z) + g(z+1)$$

## SVD

- Mit diesem Verfahren kann man Gleichungen der Form lösen. Dabei wird die Matrix A in drei Matrizen aufgeteilt, die zusammen Multipliziert wieder die Matrix A ergeben.

$$\begin{pmatrix} A \end{pmatrix} = \begin{pmatrix} U \end{pmatrix} \cdot \begin{pmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_n \end{pmatrix} \cdot \begin{pmatrix} V^T \end{pmatrix}$$

## Eigenschaften der Matrizen

$$\begin{pmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{pmatrix} U^T \cdot \begin{pmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{pmatrix} U = \begin{pmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{pmatrix} V^T \cdot \begin{pmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{pmatrix} V = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

- Wenn die drei Matrizen U, W und V einmal ermittelt sind, dann kann man z.B die Inverse der Matrix A berechnen

$$A^{-1} = V \cdot [\text{diag}(1/w_j)] \cdot U^T$$

$$x = V \cdot [\text{diag}(1/w_j)] \cdot (U^T \cdot b)$$

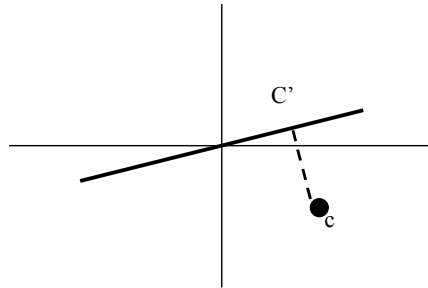
## Problemanalyse

- Bei genauerer Betrachtung fällt auf, dass die Lösung nicht definiert ist, wenn ein  $w$  den Wert Null hat. Dieser Fall entsteht, wenn das Gleichungssystem unlösbar ist. Mit Hilfe der  $w$ 's kann man eine Matrix analysieren und das „Problem“ der Matrix feststellen.

- Wenn das Gleichungssystem aber unlösbar ist, entsteht ein Fehler. Das SVD-Verfahren kann aber eine Lösung finden, die diesen Fehler minimiert.
- So kann man ein Überbestimmtes Gleichungssystem mit Ungenauigkeiten nehmen und eine optimale Lösung berechnen.

$$r = |A \cdot x - b|$$

## Veranschaulichung



## Die mathematische Problemstellung

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

$$\mathcal{O} = \sum_{i=1}^N \sum_{j=1}^P \{w(Z_{ij}) [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 +$$

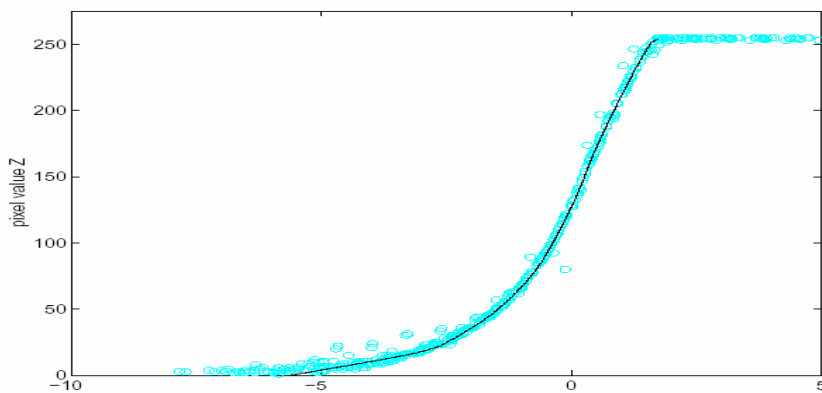
$$\lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2$$

## Konstruktion einer High Dynamic Range Map

$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j$$

$$\ln E_i = \frac{\sum_{j=1}^P w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(Z_{ij})}$$

## Ermittlung der nicht linearen Abbildung



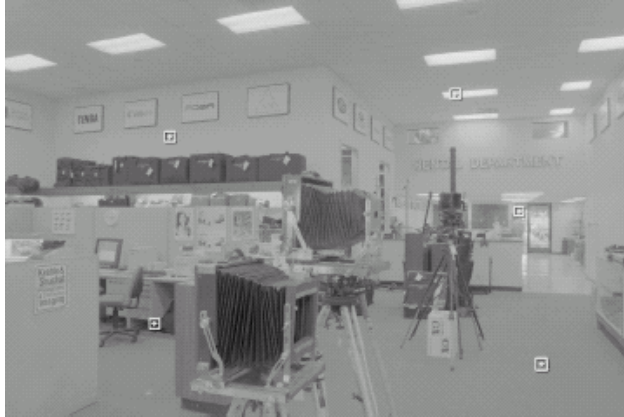
## Mehrere Bilder mit unterschiedlichen Beleuchtungszeiten



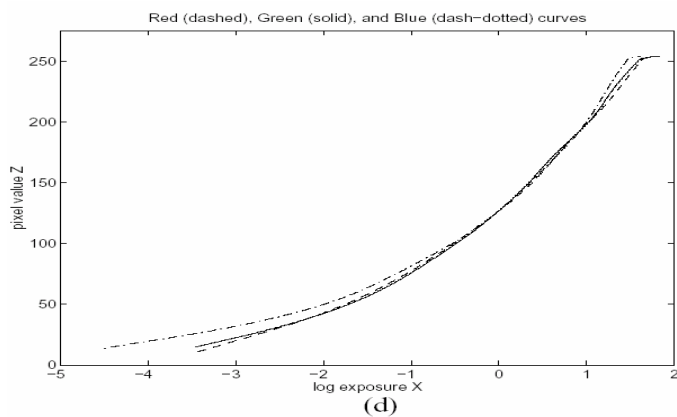
## Geringe Dynamik in den Einzelbildern



# Abbildung der rekonstruierten High Dynamic Range Map



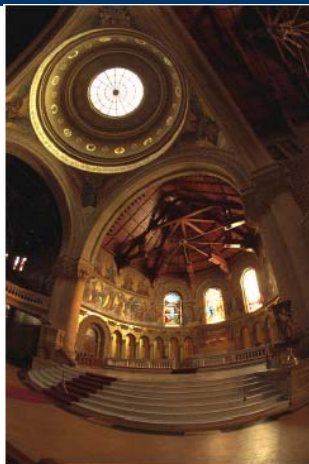
# Response functions der drei Farbkanäle Rot, Grün und Blau



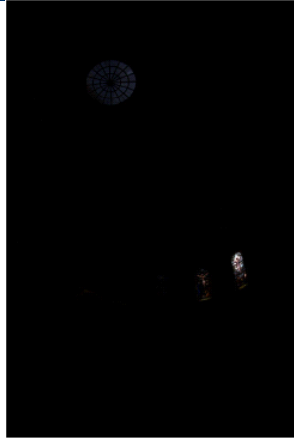
## „16“ Bilder in einer Kriche



## Echtes Foto der Kirche



## Lineare Abbildung der HDRMap in den darstellbaren Bereich des Ausgabemediums



## Lineare Abbildung der unteren 0,1% des Wertebereiches der HDRMap



## Darstellung der HDRMap durch adaptive Histogramm Kompression



## Adaptive Histogramm Kompression mit Berücksichtigung der psychologischen Wahrnehmung



## Synthetisches Motionblur vs. echtem Motionblur



## HRD Motionblur vs. echtem Motionblur



# Mathlab

```
function [g,lE]=gsolve(Z,B,l,w)
n = 256;
A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);
%% Include the data-fitting equations
k = 1;
for i=1:size(Z,1)
    for j=1:size(Z,2)
        wij = w(Z(1,j)+1);
        A(k,Z(1,j)+1) = wij;  A(k,n+1) = -wij;    b(k,1) = wij * B(1,j);
        k=k+1;
    end
end
%% Fix the curve by setting its middle value to 0
A(k,129) = 1;
k=k+1;
%% Include the smoothness equations
for i=1:n-2
    A(k,1)=1*w(i+1);    A(k,i+1)=-2*1*w(i+1);  A(k,i+2)=1*w(i+1);
    k=k+1;
end
%% Solve the system using SVD
x = A\b;
g = x(1:n);
lE = x(n+1:size(x,1));
```